

PROBABILISTIC COUNTER

BACKGROUND OF THE INVENTION5 A. Field of the Invention

The present invention relates generally to counters and, more particularly, to counters designed to count a large number of items.

10 B. Description of Related Art

Conventional networks typically include routers that route packets from one or more sources to one or more destinations. A packet is a variable size record that is transmitted through a network. A router is a network device that receives packets containing data and control information at input ports, and, based on destination or other information included in the packets, routes the packets to appropriate output ports that lead to either the next router in the packet's journey or to the packet's final destination. Routers determine the proper output port for a particular packet by evaluating header information included in the packet.

20 Routers have the capability to drop packets in a controlled fashion. This facility is used by network operators to control the amount of traffic entering or leaving a network on a given port. For diagnostic and record keeping purposes, it may be desirable for the router to keep a running total of the number of dropped packets, and the total number of bytes in the packets. In modern, high-
25 performance routers, which may process millions of packets per second, the number of dropped packets can accrue quickly. Keeping track of the total

number of dropped packets, and especially the total number of bytes dropped, may thus require large hardware registers (e.g., 64 bits or more) that can be time consuming to update.

Thus, there is a need in the art to be able to efficiently keep a counter
5 capable of quickly counting a large number of items.

SUMMARY OF THE INVENTION

Systems and methods consistent with the present invention address this and other needs by providing a high-speed probabilistic counter.

10 More particularly, one aspect of the present invention is directed to a method of probabilistically counting a series of items. The method includes generating a random number in a range defined by a first value. The random number is used to determine if the count value should be increased. Specifically,
15 the count value is increased when the generated random number is less than a second value. Finally, the range of the generated random number is increased when the count value has been increased a predetermined amount.

Another aspect of the present invention is directed to a counter for probabilistically counting a plurality of items. The counter includes a random number generator configured to generate a random number in a range defined
20 by a first value and a count value register for holding a representation of a present count of the items. The count value register includes a mantissa portion and an exponent portion. An adder is connected to the count value register. The adder increments the portion of the count value register that holds the mantissa

when the generated random number is less than a value based on at least one of the items. The adder increments the portion of the count value register that holds the exponent when the portion that holds the mantissa overflows. An update component recalculates the first value as a higher value when the
5 exponent is increased by the adder.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, explain the invention. In the drawings,

10 Fig. 1 is a diagram illustrating, at a high-level, functional components of an exemplary network router;

Fig. 2 is a graph illustrating the operation of a rate-limiter in the router shown in Fig. 1;

Fig. 3 is an exemplary block diagram illustrating functional relationships between components in a dropped packet counter in the rate-limiter;

15 Fig. 4 is a flow chart illustrating methods consistent with the present invention for operating the dropped packet counter.

DETAILED DESCRIPTION

The following detailed description of the invention refers to the accompanying drawings. The same reference numbers in different drawings identify the same or similar elements. Also, the following detailed description

does not limit the invention. Instead, the scope of the invention is defined by the appended claims and equivalents.

As described herein, a probabilistic counter includes a storage register divided into an exponent and a mantissa. The counter is probabilistically
5 incremented based on the current count. The probability of incrementing the counter decreases as the count of the counter increases. The counter has a large dynamic range and approximately constant precision relative to the number of counted items. The counter is easily updated and can be implemented using a relatively small storage register.

10 The statistically based counter consistent with the present invention may be implemented in the context of a network device, such as a router. Fig. 1 is a diagram illustrating, at a high-level, functional components of an exemplary router 100. In general, router 100 receives incoming packets 110, determines the next destination (the next “hop” in the network) for the packets, and outputs
15 the packets as outbound packets 111 on physical links that lead to the next destination. In this manner, packets “hop” from router to router in a network until reaching their final destination.

Router 100 includes routing engine 105 and a packet forwarding engine (PFE) 106. Routing engine 105 may maintain one or more routing tables (RTs)
20 115 and a forwarding table (FT) 116. Through routing tables 115, routing engine 105 consolidates routing information that the routing engine learns from the routing protocols of the network. From this routing information, the routing protocol process may determine the active routes to network destinations and

install these routes into forwarding table 116. Packet forwarding engine 106 may consult forwarding table 116 when determining the next destination for incoming packets 110.

If more incoming packets 110 are received at router 100 then the router
5 can handle or is configured to accept, it may “drop” some of the packets. For a dropped packet to successfully arrive at its destination, it must be retransmitted by its transmitting source.

Router 100 includes a rate-limiter 120 that drops packets when the input
bandwidth of the input packets 110 is larger than what the router has been
10 configured to accept. Fig. 2 is a graph illustrating the operation of rate-limiter 120. As shown, rate-limiter 120 operates to limit the output packet bandwidth when the input bandwidth is greater than a threshold value, T. The threshold may be set based on desired operating parameters. Below the threshold, the rate-limiter 120 allows all of the incoming packets 110 to pass.

15 Returning to Fig. 1, rate-limiter 120 may include storage elements, such as registers 121 and 122, that store information relating to dropped packets, such as the number of dropped packets or the total number of bytes of the dropped packets. This information is updated in registers 121 and 122 by the dropped packet counter 125 (described in more detail below).

20 Registers 121 and 122 may each be 32 bits in size. A 32-bit register, if interpreted as a single unsigned number, can count as high as about 4.3 billion. Values higher than this will overflow the register. As described below, however,

dropped packet counter 125 updates register 121 and 122 in a manner such that overflow is not an issue.

Dropped packet counter 125 will now be described in more detail with reference to Figs. 3 and 4.

5 Fig. 3 is an exemplary block diagram illustrating functional relationships between components in dropped packet counter 125. As shown, dropped packet counter 125 includes a packet size storage register (L) 303, a units variable storage register (U) 304, and a random number storage register (R) 302. A random number generator 301 supplies random number register 302 with
 10 random numbers. A comparator 306 receives values from random number register 302 and packet size register 303. An adder, under the control of comparator 306, increments a count value register 330, which may be, for example, register 121 or 122. Based on the value in count value register 330, update component 308 updates unit variable register 304.

15 Count value register 330 contains the running total of the count and may be divided into two sections, exponent section 331 and mantissa section 332. If count value register 330 is a 32-bit register, exponent section 331 and mantissa section 332 may be, for example, 6 bits and 26 bits in length, respectively. The value stored in register 330 is an estimation of the true count value of the number
 20 of total dropped packets. The estimate of the count value may be defined as:

$$2^n + m \bullet 2^{n-N_m},$$

where m is the value in mantissa section 332, n is the value in exponent section 331, and N_m is the length, in bits, of the mantissa section 332.

The operation of dropped packet counter 125 will be further described with reference to the flow chart of Fig. 4.

To begin, for each dropped packet, the length of the packet, L , is received by packet size register (L) 303. (Act 401). Random number generator 301
 5 generates a random number, R , between zero and the value of the units variable, U , that is stored in units variable register 304. (Act 402). Comparator 306 determines whether L is greater than R . (Act 403). If L is not greater, the algorithm terminates for this iteration (Act 404). If L is greater than R , comparator 306 instructs adder 307 to increment the mantissa 332 of count value
 10 register 330 by one. (Act 405).

Eventually, mantissa section 332 may reach its maximum value. If mantissa section 332 is, for example, 26 bits, its maximum value is 67,108,863. Incrementing mantissa section 332 beyond this point will cause it to overflow. In particular, if one is added to the mantissa section 332 when it is at its maximum
 15 value, it will reset to zero. When this overflow condition occurs, adder 307 adds one to exponent section 331. (Acts 406 and 407). In practice, adder 307 may treat all of count value register 330 as a single unit. In this situation, resetting mantissa section 332 to zero and adding one to exponent section 331 all occur as a natural consequence of adding one to count value register 330.

20 When the exponent section is increased, update component 308 recalculates and stores the updated value in unit variable register 304. U may be calculated based on the value in exponent section 331 (n), and on the length (N_m) of the mantissa section 332, as:

$$U = 2^{n-N_m}.$$

In this manner, U is initially near zero and is steadily increased as the count value increases. The probability of counting a particular packet at any particular time is L/U .

5 The functional implementation of dropped packet counter 125, as shown in Fig. 3, is exemplary only. One of ordinary skill in the art will recognize that dropped packet counter 125 could be implemented with a variety of different components and configurations.

10 Dropped packet counter 125, as described above, counts the number of dropped packet bytes using a probabilistic counting scheme. Because every dropped packet is not explicitly added to the count value register 330, the required update speed of count value register 330 can be less than the update speed if every dropped packet were counted, while still maintaining a constant precision as a percent of the count value. Further, because the count value is
15 represented using an exponent and mantissa representation, the dynamic range of the counter can be very large. Finally, the implementation of dropped packet counter 125 is relatively simple, using only a random number generator, storage registers, a comparator, an adder, and a numerical calculating section.

20 Although described in the context of a counter counting dropped packets for a router, the concepts consistent with the present invention are not limited to implementation within a router. The dropped packet counter 125 can be generalized to probabilistically count any set of items.

The foregoing description of preferred embodiments of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired
5 from practice of the invention.

Although described as being primarily implemented in hardware, the present invention may be embodied in hardware and/or in software (including firmware, resident software, micro-code, etc.). Furthermore, the present invention may take the form of a computer program product on a computer-usable or
10 computer-readable storage medium having computer-usable or computer-readable program code embodied in the medium for use by or in connection with an instruction execution system.

The scope of the invention is defined by the claims and their equivalents.